

FOS CDR RID Report

Date Last Modified 12/14/95

Originator M. Kikucki

Organization JAROS

E Mail Address jaros@sml.co.jp

Document FOS CDR Day 3 - Volume 1

Phone No -8-1-5543-1061
(Japan)

RID ID	CDR 15
Review	FOS
Originator Ref	
Priority	2

Section RMA

Page BN-7

Figure Table

Category Name Quality

Actionee ECS

Sub Category

Subject Software Availability

Description of Problem or Suggestion:

FOS Hardware RMA is excellent. However, I have a concern that software availability = 1.0 for analysis even though software and hardware will be measured during system test and operation.

It is necessary to evaluate the RMA for software programming you developed by some quality control technique - e.g., software bug control, software version control, etc.

Originator's Recommendation

GSFC Response by:

GSFC Response Date

HAIS Response by: Andy Miller

HAIS Schedule

HAIS R. E. Scott Carter

HAIS Response Date 11/10/95

A statistical reliability estimate of custom code during the design phase is not feasible due to the unavailability of software test or operational data. All software reliability estimation models, i.e. Schneidewind, Jelinski/Moranda, Musa/Okumoto, Littlewood/Verrall to name a few, are time execution models which require extensive data collection as input parameters.

However, FOS software reliability is ensured via a formal software development process, extensive software inspections and code walkthroughs, and a very stringent software test program at every level of the software design process.

Several tools have been identified/acquired to support software quality control techniques. Tools such as the McCabe quality Tool Set Standard will allow FOS software developers to explicitly identify high risk areas with both metrics and visual inspection within the FOS architecture. This tool also gives software developers instantaneous testing, verification, integration and shows discrepancies between design and code. Thus, this will minimize or remove all software residual errors after testing.

Additionally, there are tools that have been identified to address bug control (via DDTS tracking) and software version control (Clearcase).

DDTS is the Distributed Defect Tracking System which was developed by QualTrak for use by software, hardware, and quality labs and customer support. DDTS is project based. Bugs can be either software or hardware and may be logged at any system on the network. Bug reports are routed automatically, and submitters are informed of the status of their bugs via a graphical user interface.

Status Closed

Date Closed 12/14/95

Sponsor Johns

***** Attachment if any *****